Amendments to the Specification

1. Please replace the TITLE with the following:

A SPECULATIVE BRANCH TARGET ADDRESS CACHE THAT OUTPUTS THE TARGET ADDRESS OF A BRANCH INSTRUCTION AND SPECULATIVE BRANCH INFORMATION IN RESPONSE TO A FETCH ADDRESS WHETHER OR NOT THE BRANCH IS PRESENT IN THE INSTRUCTION CACHE LINE INDEXED BY THAT ADDRESS.

2. Please amend the table on page 1 of the disclosure as follows:

Docket #	Serial #	Title
CNTR:2022	09/849658	APPARATUS, SYSTEM AND METHOD FOR DETECTING AND CORRECTING ERRONEOUS SPECULATIVE BRANCH TARGET ADDRESS CACHE BRANCHES
CNTR:2023	09/849734	SPECULATIVE HYBRID BRANCH DIRECTION PREDICTOR
CNTR:2050	09/849822	DUAL CALL/RETURN STACK BRANCH PREDICTION SYSTEM
CNTR:2052	09/849799	SPECULATIVE BRANCH TARGET ADDRESS CACHE WITH SELECTIVE OVERRIDE BY SECONDARY PREDICTOR BASED ON BRANCH INSTRUCTION TYPE
CNTR:2062	09/849754	APPARATUS AND METHOD FOR SELECTING ONE OF MULTIPLE TARGET ADDRESSES STORED IN A SPECULATIVE BRANCH TARGET ADDRESS CACHE PER INSTRUCTION CACHE LINE
CNTR:2063	09/849800	APPARATUS AND METHOD FOR TARGET ADDRESS REPLACEMENT IN SPECULATIVE BRANCH TARGET ADDRESS CACHE

3. Please replace the SUMMARY, which begins at page 14, with the following:

BRIEF SUMMARY OF THE INVENTION

The present invention provides a branch prediction method and apparatus that makes efficient use of chip real estate, but also provides accurate branching early in the pipeline to reduce branch penalty. Accordingly, in attainment of the aforementioned object, it is a feature of the present invention to provide a branch target address cache (BTAC) for providing a speculative target address to address selection logic, the address selection logic selecting a fetch address for addressing a line in an instruction cache, the BTAC providing the speculative target address based on a presumption that a branch instruction is present in the cache line. The BTAC includes an array of storage elements, configured to cache target addresses of previously executed branch instructions and to store speculative branch information associated with the previously executed branch instructions, wherein the speculative branch information comprises a length of the branch instruction presumed present in the cache line. The BTAC also includes an input, coupled to the array, for receiving the fetch address, to index into the array of storage elements to select one of the target addresses. The BTAC also includes an output, coupled to the array, for providing the one of the target addresses indexed by the fetch address to the address selection logic. The output provides the one of the target addresses to the address selection logic for selection as a subsequent fetch address whether or not a branch instruction is present in the line of the instruction cache addressed by the fetch address.

In another aspect, it is a feature of the present invention to provide a pipelined microprocessor. The microprocessor includes an instruction cache that is indexed by a fetch address, the instruction cache for caching instructions, and for providing the instructions to an instruction buffer. The microprocessor also includes a branch target address cache, coupled to the instruction buffer and indexed by the fetch address, for caching branch target addresses of previously executed branch instructions. The instruction buffer includes a plurality of indicators that are associated with the instructions. The indicators specify whether the microprocessor has speculatively branched to one of the branch target addresses.

In another aspect, it is a feature of the present invention to provide a method of speculatively branching in a pipelined microprocessor. The method includes caching a plurality of branch target addresses of previously executed branch instructions and an indication of whether each of the branch instructions spans an instruction cache line in a branch target address cache (BTAC). The method also includes accessing the BTAC with a fetch address of an instruction cache after the caching. The method also includes determining whether the fetch address hits in the BTAC in response to the accessing. The method also includes branching the microprocessor to one of the plurality of branch target addresses selected by the fetch address if the fetch address hits in the BTAC whether or not a branch instruction is cached in a line of the instruction cache indexed by the fetch address.

In another aspect, it is a feature of the present invention to provide a branch target address cache (BTAC) for providing a speculative target address to address selection logic, the address selection logic selecting a fetch address for addressing a line in an instruction cache, the BTAC providing the speculative target address based on a presumption that a branch instruction is present in the cache line. The BTAC includes an array of storage elements, configured to cache target addresses of previously executed branch instructions and to store speculative branch information associated with the previously executed branch instructions. The speculative branch information comprises an indication of whether the branch instruction presumed present in the cache line spans more than one line in the instruction cache. The BTAC also includes an input, coupled to the array, for receiving the fetch address, to index into the array of storage elements to select one of the target addresses. The BTAC also includes an output, coupled to the array, for providing the one of the target addresses indexed by the fetch address to the address selection logic. The output provides the one of the target addresses to the address selection logic for selection as a subsequent fetch address whether or not a branch instruction is present in the line of the instruction cache addressed by the fetch address.

In another aspect, it is a feature of the present invention to provide a method of speculatively branching in a pipelined microprocessor. The method includes caching a plurality of branch target addresses of previously executed branch instructions and a length of each of the branch instructions in a branch target address cache (BTAC). The method also includes accessing the BTAC with a fetch address of an instruction cache after the caching. The method also includes determining whether the fetch address hits in the BTAC in response to the accessing. The method also includes branching the microprocessor to one of the plurality of branch target addresses selected by the fetch address if the fetch address hits in the BTAC whether or not a branch instruction is cached in a line of the instruction cache indexed by the fetch address.

An advantage of the present invention is that because it is not integrated with the instruction cache and caches branch target addresses and prediction information only for branch instructions, it potentially makes more efficient use of integrated circuit real estate than a BTAC integrated into the instruction cache which also caches branch target addresses and prediction information for non-branch instructions.

Another advantage of the present invention is that because the BTAC is relatively small by virtue of not being integrated into the instruction cache, it is potentially more amenable to implementation as a single-cycle cache than a BTAC integrated into the instruction cache, thereby potentially enabling branching earlier than the integrated solution.

A further advantage of the present invention is that it does not require indexing of the BTAC by an instruction pointer prior to the branch instruction being predicted according to a previous method, thereby avoiding the negative impact upon prediction accuracy of the previous method.

A further advantage of the present invention is that it enables an early speculative branch without the potential branch instruction being decoded by instruction pre-decode logic to determine whether a line in the instruction cache actually contains a branch instruction.

Other features and advantages of the present invention will become apparent upon study of the remaining portions of the specification and drawings.

4. Please amend paragraph 87 on page 37 of the disclosure as follows:

The non-speculative branch direction predictor 412 generates a non-speculative prediction of the direction of a branch instruction 444, i.e., whether the branch will be taken or not taken, in response to the instruction decode information 492 received from the instruction decode logic 436. Preferably, the non-speculative branch direction predictor 412 includes one or more branch history tables for storing a history of resolved directions of executed branch instructions. Preferably, the branch history tables are used in conjunction with decode information of the branch instruction itself provided by the instruction decode logic 436 to predict a direction of conditional branch instructions. An exemplary embodiment of the non-speculative branch direction predictor 412 is described in U.S. Patent No. 6,550,004 entitled ~~application serial number 09/434,984 (Docket Number CNTR:1498)~~ HYBRID BRANCH PREDICTOR WITH IMPROVED SELECTOR TABLE UPDATE MECHANISM, having a common assignee and which is hereby incorporated by reference. Logic that ultimately resolves the direction of the branch instruction preferably resides in the E-stage 326 of the pipeline 300.

5. Please amend paragraph 90 on page 39 of the disclosure as follows:

An exemplary embodiment of the non-speculative call/return stack 414 is described in U.S. Patent No. 6,314,514 entitled ~~application serial number 09/271,591 (Docket Number CNTR:1500)~~ METHOD AND APPARATUS FOR CORRECTING AN INTERNAL CALL/RETURN STACK IN A MICROPROCESSOR THAT SPECULATIVELY EXECUTES CALL AND RETURN INSTRUCTIONS, having a common assignee and which is hereby incorporated by reference.

6. Please amend paragraph 91 on page 39 of the disclosure as follows:

The non-speculative target address calculator 416 generates the non-speculative target address 354 of Figure 3 in response to the instruction decode information 492 received from the instruction decode logic 436. Preferably, the non-speculative target address calculator 416 includes an arithmetic logic unit for calculating a branch target address of PC-relative or direct type branch instructions. Preferably, the arithmetic logic unit adds an instruction pointer and length of the branch instruction to a signed offset comprised in the branch instruction to calculate the target address of PC-relative type branch instructions. Preferably, the non-speculative target address calculator 416 includes a relatively small branch target buffer (BTB) for caching branch target addresses of indirect type branch instructions. An exemplary embodiment of the non-speculative target address calculator 416 is described in U.S. Patent No. 6,609,194 entitled ~~application~~

~~serial number 09/438,907 (Docket Number CNTR:1507)~~ APPARATUS FOR PERFORMING BRANCH TARGET ADDRESS CALCULATION BASED ON BRANCH TYPE, having a common assignee and which is hereby incorporated by reference.